Abrianna Johnson

9/21/25

Program description:

This program prompts the user to enter an email message and scans the message for potential spam keywords and phrases. The program then displays the number of keywords that appeared in the message, the likelihood of the message being a spam email, and the keywords that caused it to be spam.

Functions used in the program:

1. Calculate\_spam\_score(message, keywords)

Description:

The function sets an accumulator for the number of spam keywords and counts the number of times a keyword appears in the message. The function also converts the message to all lowercase so that the keywords found in the message match the list of keywords.

Parameters:

(message, keywords) - The message the user inputs and the keywords written in the program

Variables:

Spam\_score – The number of spam keywords found in the user’s message

Found\_keywords – The actual word found in the message that is spam

Count – counts the number of lowercase keywords found and gets added to spam\_score

Logical steps:

1. Create an accumulator for the score
2. Create an accumulator for keywords
3. Convert the message to all lowercase
4. Add a point to the spam score if a keyword is found in the message

Returns:

spam\_score – returns number of keywords found

found\_keywords – returns keyword found in message

1. Spam\_likelihood(score)

Description:

This function sorts the number of spam words found and assigns a phrase that describes the likeliness of the message to be spam

Parameters:

(score) – number of keywords found in the message

Variables: none

Logical steps: sort each number into a set to determine the likelihood of the message being a spam message

Returns:

Text that displays the likelihood of the message being spam to the user

1. Main()

Description:

The function prompts the user for their email message and calculates the final spam score and spam likelihood. It also displays the results and the exact words that are listed as spam keywords.

Parameters: none

Variables:

Triggered- words that appeared in the keywords list and the user’s message

Logical steps

1. Prompt the user for their message
2. Calculate the spam score
3. Sort the number for the likelihood of the message being spam
4. Print the results
5. Print the words that are in both the keywords and the user’s message
6. Call the main function